

経営情報研究

第19巻第1号(2011年10月), 57-71ページ

## 研究論文

# 認定SAを導入したプログラミング基礎の 授業実践と評価

松 永 公 廣\*・横 山 宏\*\*・佐 野 蘭 美\*

## Evaluation of a Practice of Programing Education Using Student Support Members

Kimihiro MATUNAGA Hiroshi YOKOYAMA Mayumi SANO

【要 約】経営情報学科では、大学初年次に情報教育としてプログラミング教育を実施しているが、多様な学生を教育対象としているため、学生にプログラミングという概念を理解させるには授業実践上多くの努力が必要となる。

本論文では演習型授業「プログラミング基礎」の教育目標に到達するための授業デザインと 2010 年度に実践した授業評価について示した。

授業アンケート結果より、大学入学までにプログラミングの経験がない学生であっても、「簡単な C プログラミングができるようになる」という教育目標は達成されたと考える。また「学生が自分の学習成果を認識できるようになる」という教育目標については、ある程度達成されていると考える。また授業を受けることによって卒業後の職業に対して前向きな意見を持つようになっていることが知られる。

また授業実践の工夫として導入した「学びあい」の考え方とそれをすすめるための認定 SA の仕組みもある程度有効に機能していたと考えてさしつかえない。

学生の状況を観察し、授業設計を繰り返し、学生に授業過程を明示化し、授業途中で進捗度を調整したこと、また授業設計を実現し易い授業環境（授業支援ツールや情報処理室の設備）の整備などが、その評価結果に影響したと考える。

---

\*摂南大学、\*\*大阪電気通信大学

## 1 はじめに

大学の各学部・学科は、学問領域の特質を踏まえてカリキュラムにその理念を実現している。経営情報学科では、経営と情報のセンスをバランスよく身に付けることを目標として、大学初年次にプログラミング基礎教育を実施している。しかしながらユニバーサル社会を迎え多数で多様な学生を教育対象としているため、学生にプログラミングという概念を理解させ基本的な知識を習得させるには多くの授業実践上の努力が必要となる。演習型の授業であるため学生に「できるようになった」と実感させるには、学生が独力で問題内容を理解し、演習課題を実行できるだけのプログラミング知識を学び、プログラミングとデバッグを行い、出力例と同じ結果となるまで演習を繰り返し、独力でしかもよい成績でテストに合格させることが効果的である。しかし多様な学生を前提にすると実際の授業では進度に大きな差が生じさるる授業実践上の工夫が必要となる。それは、授業開始にあたって学生の状況を推定する、授業内容、演習環境などを詳細に設計する、授業では学生の状況を細かく観察するなどを行い、学生のつまづいている点を見つけて授業進度を調整するなど、学生が学習をあきらめない環境を作ることがこれまで以上に必要である。本論文では、それらのいくつかの条件を満たすため Moodle を活用した授業支援システムを開発し課題について教員が学生とのコミュニケーションを多く取れるように工夫した授業デザインの概略を示す。実施年度は 2010 年度後期である。

## 2 プログラミングの授業設計

プログラミング教育の教育効果を向上させるには授業実践から改善に結びつきそうな条件を抽出して優先順位を定めて教育改善の計画を立てる、実践する、教育効果を評価する、という手順を毎授業、毎年繰り返すことである。これまでに授業経験より高等教育のプログラミング教育において想定している教員、学生の授業活動は以下のようなものである<sup>(1)</sup>。

「教員」：教員がおこなう活動は、「授業設計」、「授業の実施」、「学生の指導」、「授業の評価」、「授業の改善」である。教員はまず、教育内容を整理して教育目標を明確にする。次に教育方法を決め、具体的な授業案や授業で実施する課題を作成し、授業準備をおこなう。教育を支援するツール類については、教員と学生の活動を分析して必要に応じて編集する。こうした授業準備をして、授業を実施する。対面における授業では、ツールの使用方法などを説明し、学生の活動を観察する。授業における学生の状況に応じて具体的な支援も行う。授業終了後は授業を見直し、評価をおこない、次の改善案を準備する。授業中の観察や学生の状況を推定するために行なった教育内容チェック表から得たデータをもとに、教育目標の到達度や授業の見直しを分析する。そして、必要に応じて学生に事後指導をおこない、次の授業に備える。

「学生」：学生がおこなう活動は、「授業の準備」、「受講」、「授業の復習と自己診断」である。

学生は授業の最初に教育目標を確認し、プログラミングに必要な知識を学習し、演習課題のプログラムを作成する。教員が作成した演習課題の内容を読んで理解し、課題解決に必要な情報を収集して課題の解決案を立案し、プログラミングとデバッグを行う。課題の解決案を立案するために、学生は必要に応じて、教員からの支援、教科書や配布資料を参照する、他の学生にプログラミングの困難点を相談するなどの協力を得ることがある。

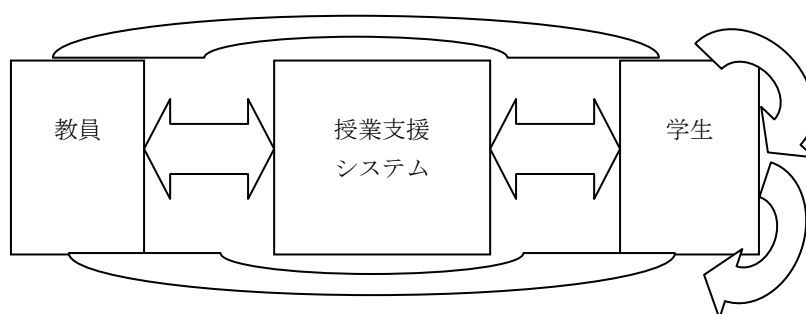


図 1 授業のモデル

「教員」、「学生」、「授業支援システム」の関係と活動を図示したのが図 1 の授業モデルである。2010 年度は学生同士の「学びあい」の仕組みを導入したことテスト直前の復習時間の設定が特徴である。

### 3 授業設計

教育目的は、C 言語の理解、プログラミング過程における思考力および実行力の育成である。

教員は助手を含めて 2 名である。教員らはこの科目を数年担当し、授業改善のために開発した独自のシステムで授業実践した<sup>(3)</sup>経験がある。

学生が社会科学系学部の 1 年生ということもあり、数や計算に対する苦手意識を持つ者が多い。そのため、やさしい多くの演習問題を考えさせて解法を理解させ、厳密で論理的な思考でない場当たりのプログラミングであれ演習に慣れさせることを目指した。

プログラミングは基本的な 1 つ 1 つのコマンドを学び、課題に対応するプログラムを作成するという演習を積み重ねなければならないが、筆者の過去の授業経験から、学生は途中でつまずくと自分でプログラミングを考えることをあきらめ教員の解答提示を待つようになるため、学生がプログラミングに取り組むことをあきらめないよう授業ごとに課題提出率やアンケートを活用して、学生の状況を把握して、授業進度を調整する。

教授戦略は、授業の最初の 30 分は授業を行い、授業で学んだ知識を確認するための授業内容確認テストを並行して行なう。残りの 60 分は 5～8 題の例題と類似した演習問題を作成して全て提出させ、授業最後に教育目標達成度のアンケートを行う。教員は各回の授業中または授業終了後、小テストの成績、課題提出率、アンケートの結果と授業時における学生の授業活動の観察から学習状況を把握する。その分析結果を次の授業に反映する。最終成績は、授業最終回に実施する演習テストの結果と各授業で提出された授業内容確認テストの成績などである。

教授戦略を遂行するには、学習状況の把握が必要であるため、LAN 環境を備えたコンピュータ室を利用し、課題や学生の学習活動を Moodle 上に実装し、教員は学習状況を把握する。

2008 年までの授業では、学生が授業中に行う学習活動（出席、小テスト、教材と課題の提示、

課題の提出、アンケート) 機能を備えた独自のシステム<sup>(2)</sup>を用いて学習状況の把握を行ない、教育改善を実施してきた。しかし、2009年より利用している Moodle (CMS) にはこれらの機能が用意されており、さらに有用な機能も備わっていること、独自システムで実装予定であった機能も用意されていることから、授業に活用する支援システムとして Moodle を用いることとした。独自のシステムで開発した機能を運用した経験が生き、Moodle のコース作成は容易に行うことができた。

2010年度から認定 SA という仕組みを導入した。学生の多様性によって授業が進むにつれて学生の進度や理解度に差が生まれることは容易に想像できる。そこで早く課題を終えた学生に SA (スチューデント・アシスタント) を依頼した。SA とは学生の教育効果を高めるために、科目担当者の補助業務に従事する学生のことである。この仕組みを導入した理由は2つある。1つ目は進度の速い学生の力をより伸ばしたいということである。教えることによってさらに深く理解させる狙いがある。2つ目はプログラミングに行き詰る学生が多くなり2名の教員では対応できない状況になり多くの補助が必要になったためである。しかし教員の増員を期待することもできないため、プログラミング教育において学生の協力を得るため「学びあい」という授業方法を取り入れることにした。学びあいを導入するにあたって「答えをおしえてはいけない、ヒントを出しなさい。そうしないと試験の時に友達が困ることになる」と繰り返し説明した。学生の多様性のため学生の学習進度に大きく差ができること逆に生かしてこのような「学びあい」「教えあい」のしくみを導入した。また友人間で相互に意味のあるコミュニケーションを経験させたいという意味もあった。

#### 4 C プログラミングツール

一般にプログラミング演習ではプログラムをコマンドラインで実行するものとしていたが、社会科学系の学生にとっては、その手順に慣れさせることよりも授業内容の学習に集中させるほうが良いと判断して、プログラムの作成から実行までをボタン操作で行なえるようにした。そのツールの実行画面を図 2 に示す。

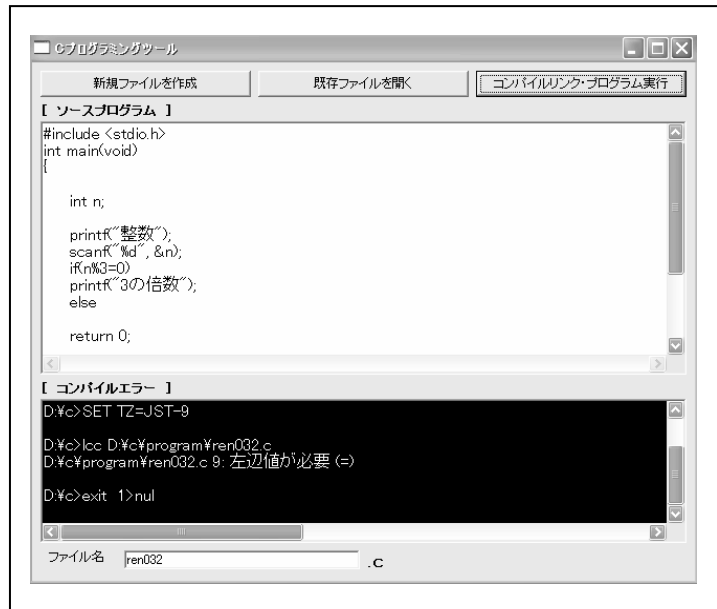


図 2 プログラミングツールの実行画面

#### 5 授業実践の評価

授業対象は、2010 年度経営学部経営情報学科 1 年生で、履修希望者は 58 名であった。最終テストを受験した学生は 53 名であった。この教科は後期配当の選択科目で、担当者は助手を含めて 2 名で、独自に開発した演習支援システムを数年にわたって実践した経歴を持つ。授業回数は演習型テストを含めて全 15 回であり、授業内容は、メッセージの表示、数の計算、キーボードからの入力、判断・分岐処理、繰り返し処理である。2010 年度の授業の到達目標を以下のように設定した。①簡単な C プログラミングができるようになる。②学生が自分の学習成果を認識できるようになる。

##### 5.1 演習テストの成績

学生の個人成績を評価するテストを授業の最終日に演習テスト形式（テスト時間は 90 分）で行った。その直前に 40 分のテスト準備の時間を設定した。授業中に使用した資料はすべて参照可とした。図 3 に授業内容確認テストの成績を含めた総括的成績評価〔演習テストの成績

(15 点満点)×5+授業内容確認テストの成績(25 点満点)の分布を示している。演習テストの内容は、メッセージの表示(1 題)、数の計算(1 題)、キーボードからの入力と判断・分岐処理(1 題)、繰り返し処理(2 題)である。問題内容は、課題文だけでヒントとなる出力例も示していない。演習テストの受験者は 53 名であった。53 人全員が合格しており、授業目標を十分に達成している。

演習テストで成績の悪かった 1 名については授業内容確認テストの成績によってかろうじて 60 点に到達した。大学の顕著なユニバーサル化により多様な学生を受入れて教育する現状では、努力をしてもテストに合格しない状況が起こるため、本授業では授業での努力を評価して授業内容確認テストの成績を加味した成績評価を導入している。将来は課題提出数も考慮することもあると考えている。努力して合格評価をえることによってほんの少しの自信が生まれることは大きな教育成果であると考えている。

## 5.2 授業アンケートの集計結果

授業設計に対して学生による授業評価を受けるため、授業の最終回に記述式のアンケートを実施した。アンケートの内容は以下の 10 項目である。

- ①小学校、中学校、高等学校などこれまでプログラミングの経験がありますか。それば何ですか。
- ②課題を読んでプログラム作成の方針をたてることができましたか。
- ③プログラム作成の方針に沿ってプログラムを作成できるようになりましたか。
- ④エラーがでても修正できるようになりましたか。
- ⑤少し難しそうな課題でも正しい答えがでるまで頑張ることができるようになりましたか。
- ⑥もし就職して急にコンピュータ部門に配属になっても何とかやれると思いますか。
- ⑦あなたがこの授業で身につけたこと(教科に関係すること関係しないこと)どんなささいなことでも教えてください。
- ⑧この授業では課題に関してアドバイスをしたり、アドバイスを受けたりを推奨しましたが、あなたは他の人にアドバイスをしましたか。
- ⑨アドバイスを受けた人に聞きます。誰からアドバイスをうけましたか。名前を教えてください。来年度の授業のやり方を考えるのに参考にします。3 人以内にしてください。ニックネームは避けてください
- ⑩来年度にむけて授業方法で改善したほうがいいと感じることを教えてください。(たとえば授業支援システム、教材、授業方法など)

このアンケートは記名式とした。

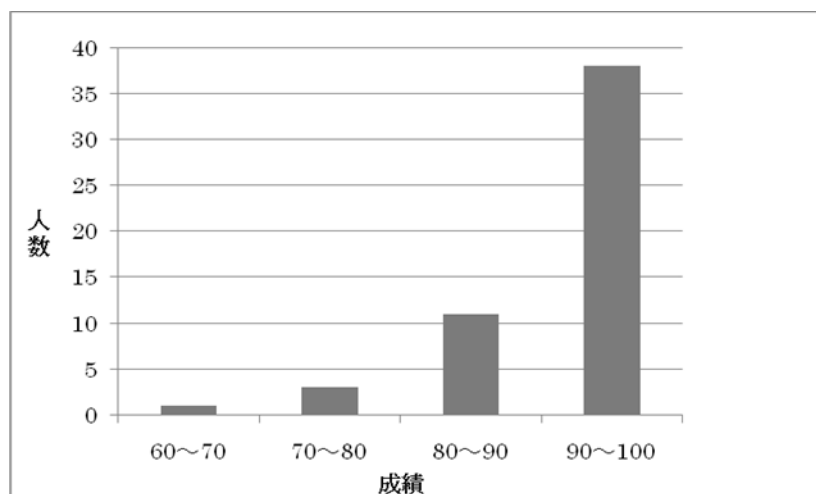


図 3 演習テストの成績

記述形式のアンケートの分析は各項目に対して全学生の記述の中から類似のキーワードをグループ化し分類する方法で行った。

アンケートの 1 項目目がプログラミングの経験の有無を聞いている。

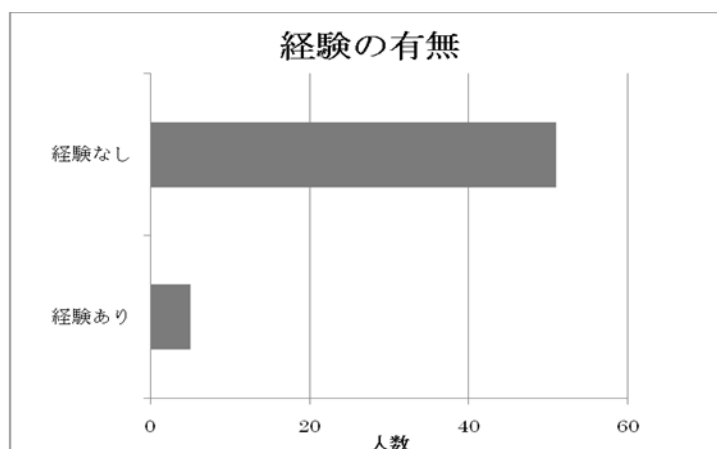


図 4 プログラムの経験の有無

大学入学までに何らかの言語でプログラムを作成した経験があるのは 53 名中 5 名であった。大多数の学生が未経験である。

アンケート②から⑤までは、プログラミングの過程を、課題を見て方針を立てる、プログラムを作成する、デバックするという 3 つの段階に分割してそれぞれの到達度を調査している。

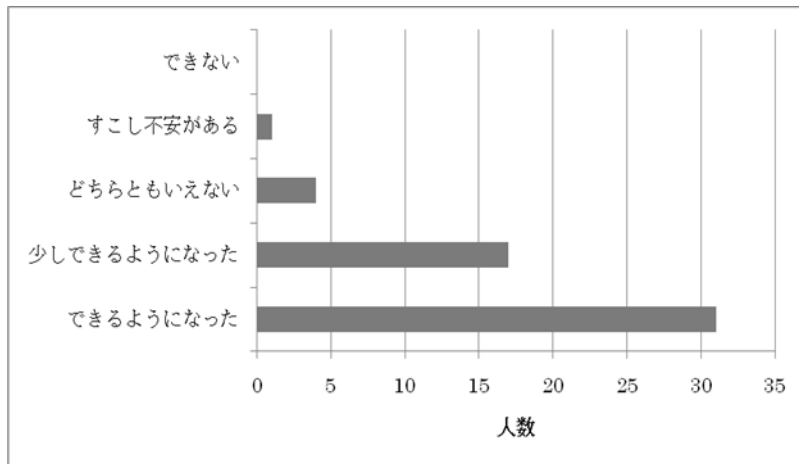


図5 プログラミングの方針をたてる

図5よりプログラミングの方針をたてるのに少し不安がある(1)、どちらともいえない(4)を含めても「方針を立てること」に不安を持つのは5名であり、他の学生は少しできるようになったと思っている。

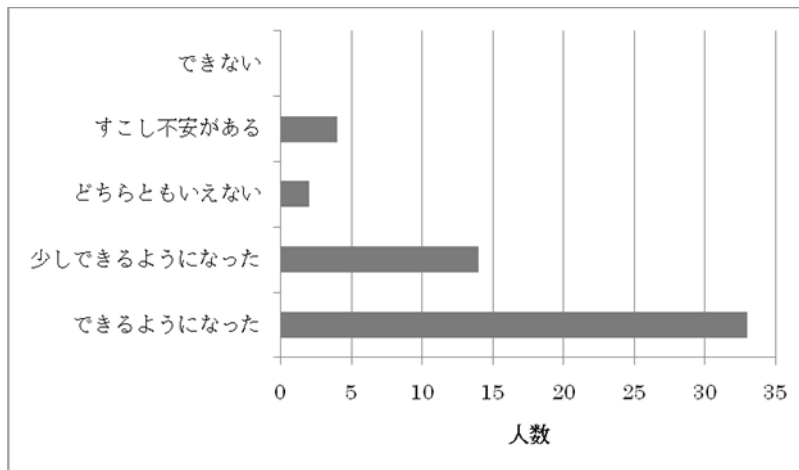


図6 プログラムを作成する

図6よりプログラミングに少し不安がある(4)、どちらともいえない(2)を含めても「プログラミングの作成に不安を持つ」学生は6名であり、他の学生はプログラミングを少しできるようになったと思っている。



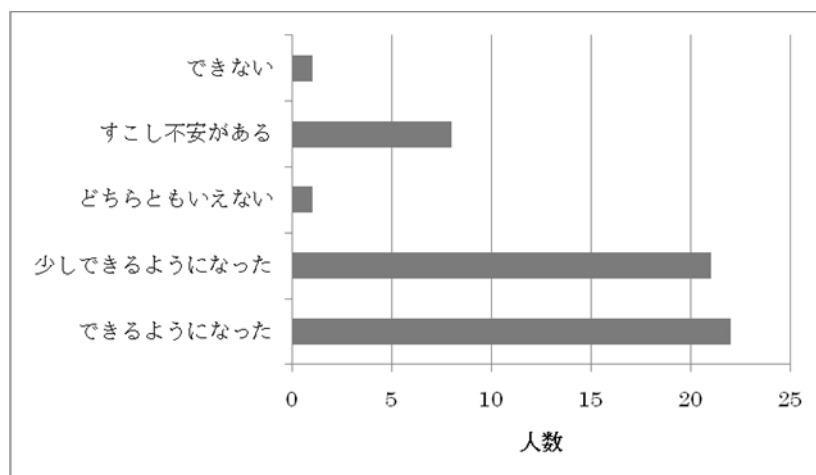


図 7 デバッグ

図 7 よりプログラムのデバッグができない (1)、少し不安がある (8)、どちらともいえない (1) を含めても「デバッグに不安がある」学生は 10 名であり、2 割程度の学生はまだプログラミングのデバッグに不安を残している。エラーを起こしても修正できないことに不安を感じているように思われる。

デバッグを正確に行うにはプログラミングのために正確な知識と実行した時に間違いを見つけ出す観察力が必要であり、慣れない学生には案外難しい。そのため学生がプログラムを作成し始めると教員に質問が集中する傾向がある。その状況を緩和するため学生の学習状況の差を生かして認定 SA を導入することにした。あとでも触れるが認定 SA はその目的を十分に果たしたことが示されている。

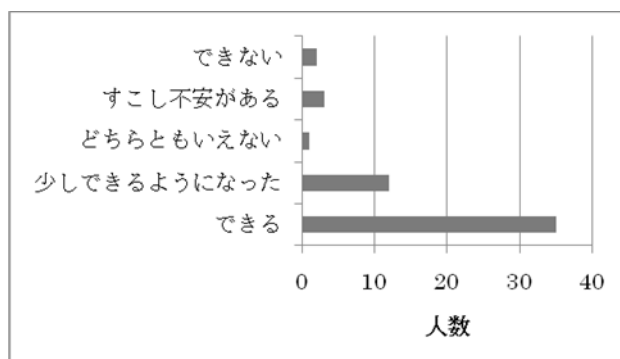


図 8 課題解決の持続力

「少し難しそうな課題でも正しい答えがでるまで頑張ることができるようになりましたか」の質問には、できない (2)、少し不安がある (3)、どちらともいえない (1) と答えた 6 名は、社会人基礎力でも触れられている「前に踏み出す力 (アクション) のうち実行力」に自信が持てない状況ではないかと考えられる。

(5) できる、(4) 少しできるようになったと回答した学生は、プログラミングを少し理解し自信が芽生えたのではないかと考えられる。社会人基礎力の「前に踏み出す力 (アクション) のうち実行力」につながったと考えたい。

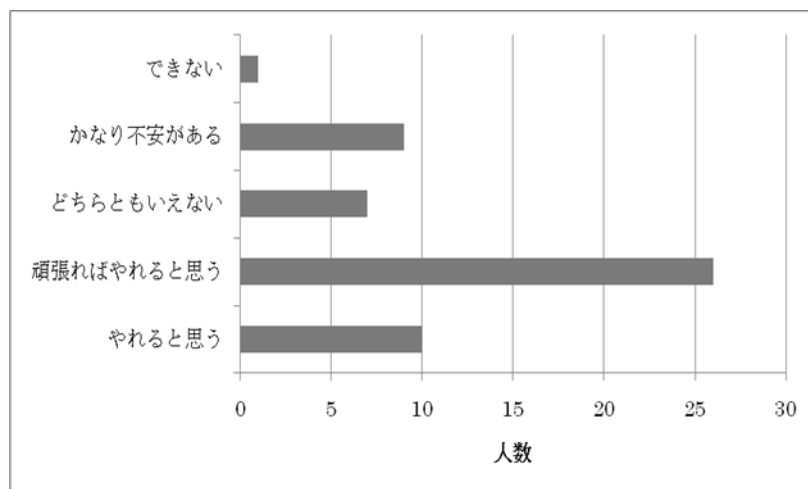


図9 コンピュータ部門への配置転換への対応

「もし就職して急にコンピュータ部門に配属になっても何とかやれると思いますか」の質問には、できない (1)、かなり不安がある (9)、どちらともいえない (7) と回答したのは 17 名であった。一方やれると思う (10)、がんばればやれると思う (26) が過半数を占めており、この授業でコンピュータ関連の仕事内容を理解し、仕事に対するアレルギーを抑えて前向きに考えられるようになっていると考えたい。

「あなたがこの授業で身につけたこと (教科に関係すること関係しないこと) どんなささいなことでも教えてください」という質問に対して記述形式の回答を収集した。図 10 に類似の記述をグルーピングした結果を示している。

プログラミングスキル、プログラミング知識、コンピュータリテラシーというキーワードに対応する記述は教科目標であるため当然であるが、プログラミング学習するなかで教育目標を越えて身につけた別の一般的な力に気づき書いている学生も多い。図 10 で他の能力として、学びあいの効果、取り組む姿勢、学び方、実行力、忍耐力、努力、課題解決、達成感、慎重さなどが挙げられている。プログラミングの学習過程で社会人基礎力で求められている一般的な能

力の習得を意識している。一方回答なしが7名おり、その7名は学習の実感がないため言葉にできないのではないかと考えられる。社会人基礎力における発信力が未成熟な状況にあるのかもしれない。今後取り組むべき課題である。

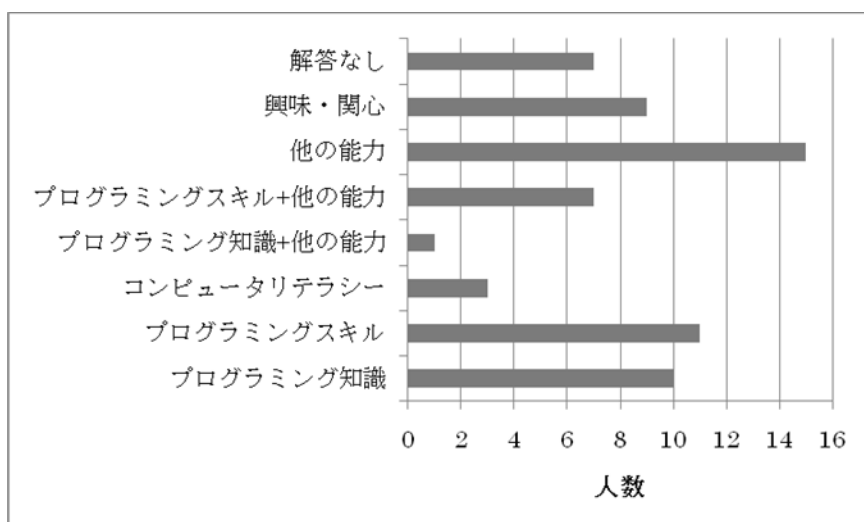


図 10 授業で見つけたこと

図 11 に「この授業では課題に関してアドバイスをしたり、アドバイスをを受けたりすることを推奨しましたが、あなたは他の人にアドバイスをしましたか」という質問の集計結果を示す。

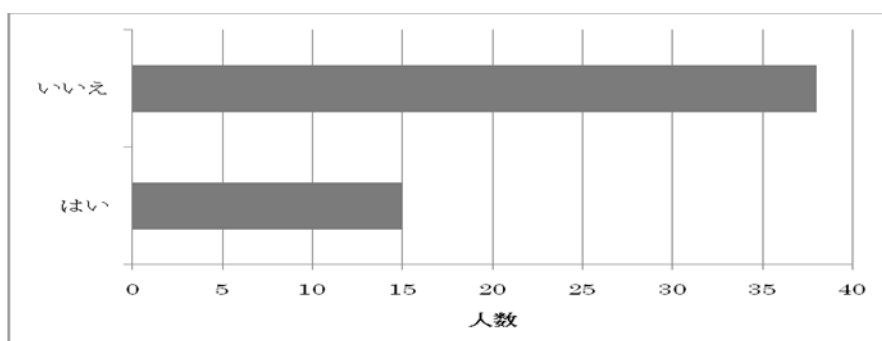


図 11 学びあいの経験

学生間での「学びあい」を取り入れた学習を推奨にも関わらず、アドバイスができるようになった人が15人であるとの結果であった。アドバイスをできる人が15人も育ったと読むか15

人しかいないと読むのか判断に苦しむ数値である。

「アドバイスを受けた人に聞きます。誰からアドバイスをうけましたか。名前を教えてください。来年度の授業のやり方を改善に参考にします。3 人以内にしてください。ニックネームは避けてください」という質問に名前が挙げた学生の名前と挙げた人の人数を表 1 に示す。

1 番目と 3 番目が認定 SA に依頼した 2 名の学生であった。他の学生も友人関係をもとに自発的にアドバイスしていることが認められる。アドバイスをうけた学生の名前を挙げない学生も 24 名いた。このうち 15 名はアドバイスをした学生であるから、アドバイスもしないしアドバイスも受けない学生数は 9 名ということになる。

図 11 と表 1 を総合してみると「学びあい」はクラスの多くの場所で人間関係を基礎に行われていると考えられる。「学びあい」を社会人基礎力の発信力や傾聴力の重要性を気づかせるきっかけとするにはさらなる工夫が求められる。

表1 アドバイスした学生

名前	投票数
A	5
B	4
C	3
D	2
E	2
F	2
G	2
H	2

表 2 は授業の改善点を整理したものである。授業方法や教材についての提案を満足させることが次年度の課題となる。特になし（24）、回答なし（13）からこの授業実施形態は学生に受け入れられていると考えられる。

表 2 改善点

分類	事例	人数
授業方法	正解表示	2
	教員数	1
	日本語表示	1
	学びあいの推進	1
	フローチャートの活用	2
	座席指定	1
教材	教科書	1
	演習課題の難度	2
	ヒント	1
	課題数	1
	教材の充実	2
学習環境	サーバの負荷	1
特になし	特になし	24
回答なし	回答なし	13

### 5.3 授業の再設計の経過

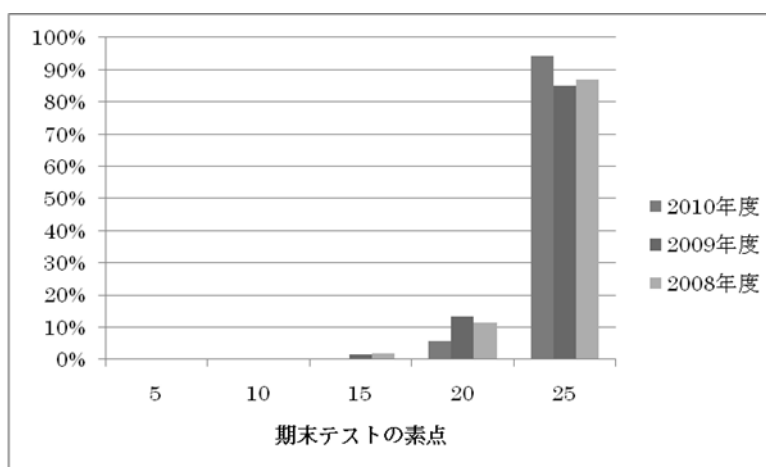


図 12 各年度の期末テストの素点の比較

図 12 に 2008 年から 2009 年の期間の最終テストの成績推移 (25 点満点) を示す<sup>(3)、(4)</sup>。2009 年度は独自開発の授業支援システムを Moodle に変更した。授業コンテンツは 2008 年度までのものを Moodle に移植した。また 2009 年度は、学生が期末テストのやり方をイメージしにくいようだと観察できたため期末試験前に模擬テストを行った。それに期末試験を受けるための総復習の役割も持たせた。

2008 年度は授業進行中に 2 つの設計変更を行った。その 1 つは、学習内容が「分岐」になったとき、学生がなかなか教育目標に到達しないことが課題の提出率と授業の観察から明らかになった。毎年分岐の概念を授業する時に苦労はしていたが、予想以上に深刻な状況であった。そのためこの授業設計の限界かとも考えたが、まず課題内容の見通しを良くし課題の内容を理解しやすいようにすることで対処しようと考え、それまで課題をパワーポイントの 1 枚のスライドにまとめていたのを、問題内容、出力例、フローチャートに分割して課題全体の見通しを良くしたところ、授業の雰囲気は改善された。学生の小さな変化を捉えて速やかに授業設計に反映することが教育目標を達成する必要条件であることを強く感じさせられた。

2 つめは、授業スケジュールの後半になって、前年まで実施していた最大値と最小値を求める問題を授業するかどうかを判断する時であった。その問題を解くにはダミー変数という新しい概念を学ぶ必要があるが、それまでの授業の観察からこれ以上新しい内容を追加することがかえって学生の理解を混乱させるリスクがあると感じられたため 2008 年度は見送ることにしたことである。大学の初年次教育であること、そして授業設計で「到達感」や「満足感」を学生に感じさせることを目標しているため、知識量を増加させるより学生の授業への満足を優先した判断であった。新しいプログラミング知識は、2 年生以降のカリキュラムの「C 言語プログラム」や「応用プログラム」が配置されているためそれらの教科に譲ることにした。

2010 年度の改善点の 1 つは授業内容の見通しを良くするため中間テストを設定したことである。中間テストの実施まえに 30 分程度の復習期間を設けた。その時間の学生の集中度は普段の授業では見られないほど高いものがあつた。そのため期末テストの前にも 40 分程度の復習時間をもうけたが、多くの学生はゆとりを持って復習していた。ある程度準備が整っていたのであろう。

さらに学習内容確認問題も 2009 年度は 87 題であつたが、1 問に複数題を含む場合があるため、学習目標が明確なるように 1 問 1 題として 193 題に分割した。その結果正解率は、2009 年度 71%が 2010 年度は 82%に向上した。

またプログラミング学習に「学びあい」という仕組みを導入した。「学びあい」を浸透させるために認定 SA という仕組みも同時に導入した。その結果「学びあい」、「認定 SA」制度が学生の演習にある程度効果であったことが知られた。

このような地道な授業改善が図 12 のような成績分布となつていると考えられる。2008 年度から 2010 年度までの期末テストの問題の難易度レベルは変わらないように作成したため信頼性は高いと考えられる。

## 6 おわりに

本論文では演習型授業「プログラミング基礎」の授業デザインを示すとともに 2010 年度にそのデザインで実践した授業評価について示した。

大学入学までにプログラミングの経験がない学生であっても授業アンケート①から④の結果より、「簡単な C プログラミングができるようになる」という到達目標は達成されたと考える。また「学生が自分の学習成果を認識できるようになる」という到達目標については、授業アンケート⑦の結果よりある程度達成されていると考える。また授業アンケート⑤と⑥より授業を受けることによって職業に対して前向きな意見を持つようになっていくことが知られる。

本論文で取り入れた「学びあい」と認定 SA の仕組みもある程度有効に機能していたと考えてさしつかえない。

それらの結果は、学生の状況を観察し、授業設計を繰り返し、学生に授業過程を明示化し、授業途中であっても進度を調整したこと、また授業設計を実現し易い授業環境（授業支援ツールや情報処理室の設備）の整備などが、学生の「授業参加」の促進に作用したと考える。

しかし学生のユニバーサル化や大学に対する社会からの要請が大きくなっていくなかで学生の能力を向上させるには、これまで以上に授業実施に関する知見を収集・分類・蓄積し、授業支援システムを授業デザインにあわせて改善して、周到に計算された授業を設計・実施・評価・改善することであろう。例えば、毎時間提出される課題のチェックは教員の大きな負担となるため処理が遅れがちになるが、その処理を支援する機能を開発し教員の負担を軽減する。またその結果を学生に個別にフィードバックすれば学生に自己の状況を理解させ、学生の授業参加を勧めることが期待される。しかしそれらのプランについても 1 つずつ授業設計に取り入れ実践によってその効果を検証していく必要がある。

このような授業実践には教員の負担増が避けられないが、得られる成果とのバランスで実施するかいなかを判断せざるを得ないであろう。

## 参考文献

- (1) 横山宏、下倉雅行、佐野繭美、松永公廣：「大学における情報教育での科目デザイン」、大阪電気通信大学人間科学研究』第 9 号、pp.15-36、2007
- (2) 佐野繭美、橋本はる美、高橋参吉、松永公廣：教育実践による C プログラミング授業方法の比較、教育システム情報学会誌、Vol.23, No.2, pp.76-82, 2006
- (3) 松永公廣、平澤洋一、情報社会における接続可能な教育改善のための CMS 活用の研究、平成 22 年度 情報文化学会第 18 回全国大会論文集、pp.19～22、2010
- (4) 松永公廣：社会系大学の情報教育におけるプログラミング教育改善のデザインと評価、平成 21 年度全国大学 IT 活用教育方法研究発表会論文集、2009